"Express Mail" mailing label number EV327128/233US

Date of Deposit: August 13, 2003

Our Case No.11178-10

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTOR:	**ARYAN SAÉD**
TITLE:	**ADAPTIVE PREDISTORTION FOR A TRANSMIT SYSTEM WITH GAIN, PHASE, AND DELAY ADJUSTMENTS**
ATTORNEY:	**CRAIG A. SUMMERFIELD BRINKS HOFER GILSON & LIONE P.O. BOX 10395 CHICAGO, ILLINOIS 60610 (312) 321-4218**

ADAPTIVE PREDISTORTION FOR A TRANSMIT SYSTEM WITH GAIN, PHASE AND DELAY ADJUSTMENTS

[00001] The present application is a continuation-in-part of United States patent application Serial No. 10/613,856 filed July 3, 2003.

Field of the Invention

[00002] The present invention relates generally to power amplification systems and is specifically applicable but not limited to power amplification systems using a Chireix architecture.

Background to the Invention

[00003] The recent revolution in communications has caused a renewed focus on wireless technology based products. Mobile telephones, handheld computers, and other devices now seamlessly communicate using wireless technology. One component that forms the core of such technology is the amplifier. Wireless devices require high efficiency amplifiers to not only extend the range of their coverage but also to conserve the limited battery power that such devices carry.

[00004] One possible architecture which may be used for such a power amplifier is called a Chireix architecture. Named after Henry Chireix who first proposed such an architecture in the 1930s, the Chireix architecture has fallen out of favor due to its seemingly inherent limitations. However, it has recently been revisited as it provides some advantages that other architectures do not have.

[00005] While the Chireix architecture provides some advantages, the process which the input signal undergoes also introduces some drawbacks. Specifically, distortions are introduced into the signal by the components in the Chireix based amplifier/modulator system. These distortions may also change over time and may therefore lead to a time-varying "drift" or change in the signal. Such distortions, time-varying or not, have led to problems that are not only inconvenient but expensive as well.

[00006] Based on the above, there is therefore a need for an amplifier system which provides the benefits of a Chireix based amplifier but which also compensates for or avoids the distortions which a Chireix based amplifier introduces. Such an amplifier system should adjust to differing conditions, preferably with little or no user intervention. It is therefore an object of the present invention to provide alternatives which mitigate if not overcome the disadvantages of the prior art.

Summary of the Invention

[00007] The present invention provides systems and methods relating to the provision of gain, phase and delay adjustments to signals to be used by a predistortion subsystem. A portion of an input signal is delayed by delay elements prior to being received by the predistortion subsystem. The delayed input signal portion is also received by a feedback signal processing subsystem that adjusts the gain and phase of the feedback signal based on the delayed input signal portion. The adjusted feedback signal is used, along with the delayed portion of the input signal, to determine an appropriate predistortion modification to be applied to the input signal.

[00008] In a first aspect, the present invention provides a system for processing an input signal, the system comprising:

- an adaptive predistortion subsystem for receiving said input signal and for producing a predistorted signal by applying a deliberate predistortion to said input signal;

- a signal processing subsystem receiving and processing said predistorted signal and producing a system output signal;

- a feedback signal processing subsystem for receiving and processing a feedback signal derived from said system output signal; and

- a delay subsystem for providing a delay to a replica of said input signal to produce a delayed signal, said delayed signal being used by said adaptive predistortion subsystem and said feedback processing subsystem,

wherein

- said adaptive predistortion subsystem distorts said input signal to compensate for distortions in said system output signal;

- said signal processing subsystem decomposes said predistorted signal into separate components, each of said separate components being processed separately;

- said processing subsystem combines said components after processing to produce said system output signal;

- an output of said feedback processing subsystem being used by said adaptive predistortion subsystem;

- said deliberate predistortion applied to said input signal by said adaptive predistortion subsystem to produce said predistorted signal is adjusted based on said system output signal;

- said predistortion subsystem is controlled by a preprogrammed control device.

[00009] In a second aspect the present invention provides a method of processing an input signal to produce a system output signal, the method comprising:

a) receiving said input signal;

b) applying a deliberate predistortion to said input signal to result in a predistorted signal;

c) decomposing said predistorted signal into at least two component signals;

d) combining said at least two component signals to produce said system output signal;

e) adjusting said deliberate predistortion based on said system output signal

f) concurrent with steps b) -e), executing the following steps:

f1) delaying a replica of said input signal;

f2) diverting a diverted replica of said system output signal to a feedback path;

f3) processing said diverted replica to produce a corrected version of said diverted replica; and

f4) determining said deliberate predistortion using said corrected version and said replica of said input signal.

[00010] In a third aspect the present invention provides an adaptive predistortion subsystem for use with a signal processing system which produces a system output signal, the predistortion subsystem comprising:

-determining means for determining a deliberate predistortion to be applied to an input signal;

- adjustment means for applying said deliberate predistortion to said input signal;

- update means for periodically updating said determining means based on characteristics of said system output signal;

- preprogrammed control means for controlling said system;

wherein said adaptive predistortion subsystem uses

- an output of a delay subsystem for delaying said input signal and

- an output of a feedback processing subsystem for processing a feedback signal derived from said system output signal to determine said deliberate predistortion.

[00011]_____In a fourth aspect the present invention provides a method of initializing a phase correction to be applied to a feedback signal, said feedback signal to be used in determining a deliberate predistortion for a signal processing system, the method comprising:

a) initiating a coarse delay search

b) selecting a time window of W samples of said feedback signal and an input signal with a predetermined sample delay increments of δ between samples

c) calculating an inner product P_δ by performing a complex multiply and accumulate process for the W samples in the time window

d) storing a maximum $|P_\delta|$ found

e) repeating steps c) and d) for subsequent time windows and incrementing δ by a predetermined amount for each time window

f) repeating steps b) - e) for a fine delay search using fractional sample increments to cover a predetermined delay range, said delay range being centered on a maximum delay increment δ_{max} found during said coarse delay search.

Brief Description of the Drawings

[00012] A better understanding of the invention will be obtained by considering the detailed description below, with reference to the following drawings in which:

Figure 1 is a block diagram of a Chireix architecture amplifier subsystem;

Figure 2 is a block diagram of an amplifier system using the subsystem of Fig 1 and a predistortion subsystem;
Figure 3 is a detailed block diagram of the internal components of the predistortion subsystem;
Figure 4 illustrates the amplifier system of Figure 2 and incorporating a feedback signal used by the predistortion subsystem of Figure 3;
Figure 5 is a block diagram of a delay line circuit which may be used in the system of Figure 4;
Figure 6 is a circuit which may be used to implement the magnitude adjustment explained below;
Figure 6A is a block diagram of a leaky integrator which may be used with a predistortion monitor illustrated in Figure 3;
Figure 6B is a state diagram illustrating the different states through which the predistortion subsystem of Figure 3 may pass through;
Figure 6C is a block diagram of one type of control block which may be used in the predistortion subsystem of Figure 4;
Figure 7 is a detailed block diagram of one embodiment of a Chireix amplifier subsystem;
Figure 8 is a block diagram of a signal processing system according to another embodiment of the invention; and
Figure 9 is a detailed block diagram of a system incorporating the feature shown in Figures 3, 4 and 8.

Detailed Description

[00013] For clarity, the following terms are to be used with the following definitions:

AM (amplitude modulation) refers to the AM of an RF (radio frequency) signal and is equal to the magnitude of the RF signal's complex base band equivalent

PM (phase modulation) refers to the PM of an RF signal and is equal to the phase of the RF signal's complex base band equivalent.

[00014] Referring to Figure 1, a block diagram of a Chireix architecture amplifier subsystem 10 is illustrated. A signal decomposer 20 receives an input complex baseband signal 30. Phase modulated signals 80A, 80B are produced after the decomposed output of the decomposer 20 are phase modulated by phase modulation circuitry 85A, 85B. These phase modulated signals 80A, 80B are received by power amplifiers 90A, 90B. The phase modulated signals are thus amplified by the power amplifiers 90A, 90B and are received by a signal combiner 100. The system output signal 110 (an RF signal corresponding to the input baseband signal 30) is output from the combiner 100 and is an amplified and modulated version of the input signal 30. Phase modulation of the phase modulated signals 80A, 80B is executed by the signal decomposer 20. The input signal 30 is separated into at least two components and these at least two components, after phase modulation, are the signals 80A, 80B.

[00015] As noted above, the Chireix architecture amplifier subsystem 10 has been known to introduce distortions in the system output signal 110. To compensate for such distortions, a predistortion subsystem 120 is provided. Referring to Figure 2, the predistortion subsystem 120 receives the input signal 30 and produces a predistorted signal 130. The predistorted signal 130 is received by the amplifier subsystem 10. The amplifier subsystem then produces the system output signal 110.

[00016] The distortions for which the predistortion subsystem is to compensate may come as a phase distortion, a magnitude distortion, or as a combination of both. It has been found that, without predistortion, the system output signal 110 has an amplitude modulation (AM) that is not equal to the expected and desired AM. Furthermore, the phase modulation (PM) of the system output signal 110, if predistortion is not present, deviates from the expected and desired PM. Experiments have found that the AM distortion or error (magnitude distortion) depends on the AM of the input signal. Also, it has been found that the PM distortion (or phase distortion) depends on the AM of the input signal.

[00017] As noted above, one solution to the above issues is to predistort the input signal as detailed in Figure 2. Further details on this approach can be found in co-pending application entitled Predistortion Circuit For a Chireix Power Amplifier Transmit System and filed with the US Patent and Trademark Office, the whole of which is incorporated herein by reference. While the predistortion solution does work, it is not as robust and as fault tolerant as may be desirable. An adaptive predistortion subsystem 200, as illustrated in Figure 3, would compensate for changing conditions and for other distortions which the system output signal may have.

[00018] Referring to Figure 3, a block diagram of such an adaptive predistortion subsystem is illustrated. The adaptive predistortion subsystem 200 of Figure 3 may be used in place of the predistortion subsystem 120 of Figure 2.

[00019] While an analog implementation of the subsystem 200 is possible, it has been found that a digital implementation was simpler to achieve. As such, the following description assumes that the input signal 30 is a digital signal having a digital representation of the desired AM and PM of the desired output RF signal. Digital AM/AM predistortion modifies the magnitude of the complex digital input signal such that the RF output signal has the desired AM, despite the distortion. Digital AM/PM predistortion modifies the phase of the complex digital input signal such that the RF output has the desired PM, despite the distortion.

[00020] As can be seen in Figure 3, multiple components are involved in the adaptive predistortion subsystem 200: a Cartesian to polar conversion unit 210, a magnitude value lookup table (LUT) block 220, a magnitude update block 230, a magnitude delay block 240, a phase value lookup table (LUT) block 250, a phase value update block 260, a phase delay block 270, and an adder 280. The digital input signal 30 is converted by the conversion unit 210 from Cartesian coordinates to polar coordinates. The magnitude of the converted signal is then received and used by the lookup table blocks 220, 250 to determine the proper amount of predistortion to be applied. The phase lookup table 250 adds the phase distortion to the converted signal by way of the adder 280. The predistorted signal is then passed on to the amplifier subsystem 10.

[00021] It should be noted that the predistortion modification, defined as any deliberate distortion which has been introduced to the input signal to change at least the phase or magnitude of the input signal, can take many forms. Figure 3, provided merely for illustrative purposes, depicts two types of predistortion - phase predistortion and magnitude predistortion. These two types, separately or together, can make up the predistortion modification. In some applications, only a magnitude type predistortion modification may be required while in others only a phase type predistortion modification is required. In the embodiment explained here, the two types of predistortion, together comprise the predistortion modification.

[00022] To account for changing conditions and to acquire appropriate LUT entries, a feedback mechanism is employed to adjust or adapt the lookup table entries in lookup table blocks 220, 250. Delay blocks 240, 270 ensure that the feedback sample is mated with the proper value of the input signal waveform when processing and updating the lookup table entries in lookup table blocks 220, 250.

[00023] The conversion unit 210, while present, is not necessary but merely convenient and makes the predistortion easier to accomplish. As is well known, signal representations using Cartesian coordinates take the form of $z = x + j\,y$ where x and y are the real and imaginary components. Polar coordinates take the form of $z = Ae^{j\phi}$ where the magnitude of the signal is A and its phase is ϕ. Since both the magnitude and the phase of the signal is to be modified by the predistortion subsystem, it is clearly more convenient to accomplish this if the signal were in polar coordinates. Again as is well known, $A = (x^2 + y^2)^{½}$ while $\phi = tan^{-1}\,(y/x)$. Once the signal has been converted into polar coordinates, adjusting the magnitude is as simple as replacing the digital representation of A by another number. Similarly, the phase can be adjusted by adding a phase correction to the phase of the signal.

[00024] After the digital input signal is received and converted by the conversion unit 210, the signal is now represented by two values - a magnitude value 290 and a phase value 300. Figure 3 shows the different signal paths followed by these values - one path for the magnitude value 290 and a second path for the phase value 300.

[00025] As noted above, the magnitude value 290 can be easily replaced by the predistorted magnitude value. This is done by way of magnitude lookup table block 220. The lookup table internal to the magnitude lookup table block 220 represents an input/output relationship with the input being the undistorted magnitude and the output being the predistorted signal magnitude. Thus, if the magnitude LUT block 220 has a table entry with an input value of 0.5 and an output value of 0.4, then if the undistorted magnitude value received by the magnitude LUT block 220 is 0.5, then this value is replaced with 0.4 as the output of the magnitude LUT block 220. Based on the LUT (lookup table) entries, the magnitude of the undistorted signal is therefore replaced with the desired predistorted magnitude.

[00026] Similar to the above, the phase value of the converted input signal is adjusted as well. As can be seen in Figure 3, the magnitude value 290 is also received by the phase lookup table block 250. The phase lookup table block 250, based on the magnitude value, determines

the proper amount of phase adjustment and adds this phase adjustment to the phase value 300 by way of the adder 280. The phase lookup table block 250 also has a lookup table resident within the phase LUT block 250 that details the appropriate phase adjustments for given magnitude values.

[00027] While the above described magnitude LUT replaces a desired value for the received magnitude, other implementations are possible. Instead of a direct replacement value, the magnitude LUT may provide a corrective value to the received magnitude. This corrective value can, depending on the implementation, be an additive or a multiplicative corrective value.

[00028] The adaptive predistortion subsystem 200 in Figure 3 is adaptive in that the values of the lookup table entries in the lookup table (LUT) blocks 220, 250 change over time to account for changing conditions or for acquiring appropriate LUT entries. This adaptability is implemented by way of a feedback signal tapped from the system output signal 110. Referring to Figures 3 and 4, two feedback signals, a magnitude feedback signal 310 and a phase feedback signal 320, are received by the magnitude value update block 230 and by the phase value update block 260 respectively. These two feedback signals result from processing of the system output signal 110 by the analog/digital (A/D) converter 330, the Cartesian to polar conversion unit 340, demodulation module 335, and filtering module 337. As can be seen in Figure 4, the system output signal (an analog signal) is tapped and this tapped signal 345 is received by the A/D converter 330 for conversion from an analog to a digital signal.

[00029] After conversion to a digital signal, the feedback signal is converted from Cartesian to polar coordinates by the conversion unit 340. The tapped signal 345 is thus represented by the two feedback signals - the magnitude feedback signal 310 and the phase feedback signal 320. As mentioned above, both these feedback signals are received by their respective update blocks 230, 260.

[00030] Once the two digital feedback signals are received, they are then compared with the delayed input signal coming from the delay blocks 240, 270. The updated values for the LUT entries are then calculated and entered into their respective lookup tables. It should be noted that the comparison may be done by subtracting the feedback signals from the delayed input signal.

[00031] To further elaborate on the above process, the update process is dependent on the difference between the tapped system output signal 345 and the input signal 30. This difference is, of course, taken after both signals are in polar coordinates. The magnitude and phase errors are defined as:

$$e_m(\mathrm{k}) = |\mathrm{z(k)}| - |\mathrm{x(k)}|$$

$$e_\phi(\mathrm{k}) = (\angle \mathrm{z(k)} - \angle \mathrm{x(k)})$$

where

$e_m(\mathrm{k})$ = magnitude error

$e_\phi(\mathrm{k})$ = phase error

$|\mathrm{z(k)}|$ = magnitude of feedback signal (signal 310)

$|\mathrm{x(k)}|$ = magnitude of input signal (signal 290)

$\angle \mathrm{z(k)}$ = phase angle of feedback signal (signal 320)

$\angle \mathrm{x(k)}$ = phase angle of input signal (signal 300)

[00032] For the magnitude LUT entries in the magnitude LUT block 220, two variables are defined and used in the process:

$$\delta_F = - \mu_F \cdot e_m(\mathrm{k})$$

where

δ_F = update quantity dependent on the differences between the magnitudes of the input signal and the feedback signal

μ_F = an update speed parameter μ (user selectable), typically $\mu_F > 0$.

[00033] Since the magnitude LUT has LUT entries, each entry is given an entry address of n with $0 \le \mathrm{n} \le -1$, N being the maximum number of entries in the internal magnitude LUT in the magnitude LUT block 220.

[00034] An interpolation distance s is defined as $s = \alpha\mathrm{M} - \mathrm{n}$ where $\mathrm{n} = \lfloor \alpha\mathrm{M} \rfloor$ (or the largest integer value less than or equal to αM), $\mathrm{M} = |\mathrm{x(k)}|$, and α is a scaling value applied such that the magnitude range (e.g. $0 \le \mathrm{M} < 1$) is mapped to a table index range $0 \le \mathrm{n} \le (-1)$.

[00035] The table entries are thus updated using the following formulae (n being one table address and n+1 being another table address) :

$$\mathrm{F_n}(\mathrm{k}+1) = \mathrm{F_n}(\mathrm{k}) + (1 - s) \cdot \delta_F \text{ iff } 0 \le \mathrm{n} \le (-1)$$

$$\mathrm{F_{n+1}}(\mathrm{k}+1) = \mathrm{F_{n+1}}(\mathrm{k}) + (s) \cdot \delta_F \text{ iff } 0 \le \mathrm{n}+1 \le (-1)$$

where

F_n (k) = table entry n for time sample k

F_n (k+1) = table entry n for time sample k + 1

F_{n+1} (k) = table entry for n+1 for time sample k

F_{n+1} (k+1) = table entry n_1 for time sample k+1

[00036] From the above equations, it should be clear that one, two or no entries in the internal magnitude LUT are updated depending on the value of αM. Thus, for this implementation, depending on the value of αM, one of F_n and F_{n+1} is updated, both are updated, or neither is updated. Other implementations, depending on the circumstances and needs of the particular application, may update other numbers of entries.

[00037] For the phase LUT entires, an analogous process is used in the phase update block 260. An update quantity is defined:

$$\delta_G = -\mu_G \cdot e_{\phi}(k)$$

where

δ_G = update quantity dependent on the differences between the phase angles of the input signal and of the system output signal

μ_G = an update speed parameter (user selectable) where $0 \leq \mu_G < 1$

[00038] Using the parameter *s* as defined above for the magnitude LUT, the phase LUT entries can be updated using the following formulas:

$$G_n(k+1) = G_n(k) + (1-s) \cdot \delta_G \text{ iff } 1 \leq n \leq (-1)$$

$$G_{n+1}(k+1) = G_{n+1}(k+1) + (s) \cdot \delta_G \text{ iff } 1 \leq n \leq N$$

[00039] Again, much like the magnitude LUT entry update process, the update process for the phase entries will, depending on the value of αM, update one, two, or none of the phase LUT entries.

[00040] It should be noted that LUT adaptation may involve updating more than two entries with some weighting applied. As an example, the weighting may depend upon the distance of the LUT entry from the update point. The discussion herein is based upon the use of two entries and the use of an interpretation distance.

[00041] As an example of the above processes, a phase entry update will be illustrated.

For this example, the following values are assumed:

$N = 6$

$M \leq 1$

$\alpha = 5$

$x(k) = 0.35 \exp (j \cdot 2 \cdot 7)$

$z(k) = 0.2 \exp (j \cdot 3 \cdot 1)$

$\mu_G = 0.1$

[00042] Thus, M=0.35 and $\alpha M = 1.75$. Thus, n=1 (since the lowest integer value less than or equal to1.75 = 1) and n+1 = 2. From these values, $s = 1.75-1 = 0.75$. Given that $z(k) = 0.2 \exp(j \cdot 2.3)$, the $e_\phi(k) = -0.4$. Thus, $\delta_G = -(0.1)(-0.4) = +0.04$. The required correction for G_1 is therefore

$(1-s) \cdot \delta_G = (1-0.75)(+0.04) = 0.25 \cdot (+0.04) = 0.01$. For G_2, the correction is $s \cdot \delta_G = (0.75)(0.04) = 0.03$. The new values are therefore:

$G_1 (k+1) = G_1(k) + 0.01$

$G_2 (k+1) = G_2 (k) + 0.03$

This update is illustrated by the values in the following table:

Address (n)	**LUT content before update**	**Correction**	**LUT content after update**
0	-1	0	-1
1	2	0.01	2.01
2	-0.5	0.03	-0.47
3	-0.5	0	-0.5
4	0.5	0	0.5
5	2	0	2

[00043] It should be noted that the above process also takes into account the lookup of values that are not to be found in the internal lookup tables. Linear interpolation using $s = \alpha M\text{-}n$ is used for magnitude values that cannot be found in the lookup table entries. As an example, if

the magnitude is given as 0.35 but the only table entries are 0.3 and 0.4, linear interpolation is used. The following formula is used to find values not found in the lookup tables:

$$F(M) = (1 - s) \cdot F_n + s - F_{n+1}$$

where

F_n = table value A

F_{n+1} = table value B with the desired value being between table values A and B

$n = \lfloor \alpha M \rfloor$ (the largest integer value less than or equal to αM)

$s = \alpha M - n$

α = scaling value as defined above.

[00044] To protect against instability under certain distortion conditions, some exceptions have to be made to the adaptation of the LUTs. The first of these is the LUT entry at address 0. The entry should be optionally frozen while allowing other LUT entries to be updated. This is done by setting (following the formula for $F_n(k+1)$)

$$F_0(k+1) = F_0(k) \text{ if } n=0$$

$$F_1(k+1) = F_1(k) + (s) \cdot \delta_T \text{ iff } n+1 = 1$$

if a freeze is desired. Such a freeze has been found to allow for added subsystem robustness if residual branch imbalances cause LUT update instability for near-zero magnitude values.

[00045] The other exception is to limit the range of amplitude values for an LUT update. By default, the magnitude range is $0 \leq M \leq 1$ but LUT updates should only occur for the magnitude range $M_0 \leq M \leq M_1$. The definition of this operative range is not dependent on the magnitude values corresponding to the LUT entries. LUT entries representing magnitude values beyond the limited range are still affected by the updates due to the use of interpolating adaptation. Limiting the active magnitude range has been found to improve stability, especially in the case of branch imbalance where phase distortion at near zero magnitude is severe. It has also been found to protect against instability for near-maximum magnitude values caused by inadvertent flipping due to pre-distortion.

[00046] From the above discussion, it should be fairly clear that two of the values underpinning most of the calculations are the values for $e_m(k)$ and $e_\phi(k)$, the magnitude and phase errors. While the equations above for these two values will provide acceptable answers, better results may be had by taking into account the phase, magnitude, and delay effects of the feedback signal (i.e. the tapped system output signal 345). However, to be able to do this, the input signal must be properly delayed so that samples from the interpolated input waveform, as obtained from the delayed input signal samples, are mated with the relevant system output signal sample. Such proper delaying should therefore take into account most, if not all, the time delay involved in the processing production, and feedback of the system output signal (round trip delay). This round trip delay is denoted as τ (seconds) and, before the LUT updating begins, the delay blocks 240, 270 acquires the delay and delays the input signal accordingly so as to time-align the input signal samples with the incoming feedback signal.

[00047] To implement this delay, a delay line is used with a depth of K, meaning K samples of the input signals may be stored in the delay line. As should be clear, each of the K samples were sampled at different instances in time. The value of K is predetermined and should be enough to allow for the maximum possible path delay between the input signal and the feedback signal. These delays are due to a combination of any of the following: digital pipelining, analog and digital filter group delays, analog propagation delays, and the system and implementation dependent delays.

[00048] Because of this delay, a time delayed version of the input signal, $x_\delta(k)$ is defined and this is ideally

$$x_\delta(k) = x(k-\delta)$$

where

$\delta = \tau \cdot \mathrm{F_s}$

$\mathrm{F_s}$ = signal sampling rate

τ = delay (normal trip delay between input and system output signal feedback)

[00049] To obtain a better result for $x_\delta(k)$, linear interpolation is used to allow for fractional values of δ. Thus, the delay is divided into two parts κ, the integer part of the sample (representing a discrete sample delay at the sample rate $\mathrm{F_s}$), and ν, a fractional sample of the delay.

[00050] Using this notation, the delayed portion of the input signal can be represented as:

$$x_\delta(k) = (1-\nu) \cdot x(k-\kappa) + \nu \cdot x(k-\kappa+1)$$

where

$$\delta = \tau \cdot F_s$$

$$\kappa = \lfloor \delta \rfloor$$

$$\nu = \delta - \kappa$$

As can be seen, for an integer δ, $x_\delta(k) = x(k-\delta)$.

[00051] The above Cartesian equation can be applied to polar representations by having separate delay lines for magnitude ($|x_\delta(k)|$) and phase ($\angle x_\delta(k)$) using the sequences $|x(k)|$ and $\angle x(k)$. These are given by:

$$\angle x_\delta(k) = (1-\nu) \cdot \angle x(k-\kappa)+\nu \cdot \angle x(k-\kappa+1)$$

$$|x_\delta(k)| = (1-\nu) \cdot |x(k-\kappa)|+\nu \cdot |x(k-\kappa+1)|$$

It should be fairly clear that $x_\delta(k)$ is calculated from the samples $x(k)$, $x(k-1)$, $x(k-2)$, ...$x(k-\kappa)$, samples of the input signal taken at time k, k-1, k-2, ... k-κ.

[00052] The above equations for $\angle x_\delta(k)$ has a peculiarity that is due to the way angle values work. Since $\angle x(k-\kappa)$ and $\angle x(k-\kappa+1)$ are represented by modulo 2π radians (360 degrees) and since $-\pi \leq \angle x(k) \leq \pi$, then errors could easily occur.

[00053] Thus, if $-\pi \leq \angle x(k) \leq \pi$, and if $|\angle x(k-\kappa+1)-\angle x(k-\kappa)| \geq \pi$, then

$$\angle x_\delta(k) = (1-\nu) \cdot \angle x(k-\kappa)+\nu \cdot (\angle x(k-\kappa+1)+2\pi) \text{ if } \angle x(k-\kappa+1) \leq \angle x(k-\kappa)$$

$$\angle x_\delta(k) = (1-\nu) \cdot \angle x(k-\kappa)+\nu \cdot (\angle x(k-\kappa+1)-2\pi) \text{ if } \angle x(k-\kappa+1) > \angle x(k-\kappa)$$

[00054] The above described delay can be implemented by cascaded delay elements and associated sample storage. Figure 5 illustrates such a delay subsystem which can be used as delay blocks 240, 270. As can be seen, delay elements 242A, 242B, 242C, 242D, 242E are cascaded and provide delays and storage for input signal samples 244A, 244B, 244C, 244D, 244E. Switches 245A, 245B, 245C, 245D, 245E allow any one of the signal samples 244A-244E to be switched so that it can be used. These samples 244A-244E can be weighted accordingly by programmable weighting blocks 246A, 246 B, 246C, 246D, 246E. The weighted samples are then summed up by adder 248 to produce the delayed signal 249 to be used by the system. The switches 245A...245E and the values in the weighting blocks 246A...246E may be user/system controllable so that any combination of weighted samples may be produced.

[00055] As an example, if $\tau \cdot F_s = 2.4$ samples is required, then a value of 0.6 is used by the weighting block 246C and a value of 0.4 is used by the weighting block 245D. Then, by closing switches 245C and 245D then the sample x(k-2.4) is obtained.

[00056] The phase correction required for the feedback portion of the system output signal is also dependent on the delayed signal x_δ(k). The complex phase difference between the delayed signal x_δ(k) and the system output signal z(k), represented by γ, is due in part to the round trip delay and to other factors. The phase correction and delay can be found by executing the following process:

1. Reset the running-maximum register: set $P_{\max} = 0$
2. Reset the phase correction: set .$\gamma = 0$
3. Preset the variable interpolating delay line (discussed above as a delay subsystem) :set $\delta = \delta_0$ (i.e. $\kappa = \kappa_0$ and $\nu = \nu_0$)
4. Perform a coarse search: select only interpolating delay line integer sample-delay increments of δ (maintain $\nu = \nu_0$, increment κ only): set the coarse and fine delay increments to $\Delta\kappa = 1$ and $\Delta\nu = 0$.
5. Start with the first correlation window $n = 0$. Time instance $k = 0$ signifies the commencement of the search.
6. "Integrate": calculate the inner product

$$P_\delta = A_{MAC} \cdot \sum_{k=n\cdot W}^{n\cdot W+W-1} \exp(j \cdot (\angle x_\delta(k) - \angle z(k)))$$

This is a complex multiply-and-accumulate (MAC) covering a time window of W current samples. Note that the above notation strictly implies that the correlation windows $n = 0,1,2,\ldots$ are adjoined, yet this is not an operational requirement. A spacing between windows translates to a lengthening of the search.

7. "Dump": compare the complex magnitude $|P_\delta|$ with the value $P_{\max}$, and if

   $|P_\delta| > P_{\max}$ then

   (a) update the maximum: set $P_{\max} = |P_\delta|$

   (b) update the delay: retrieve the corresponding value of δ and store it as $\delta_{\max}$ and

   c) update the angle: if fine searching store the angle $\gamma = \angle P_\delta$

8. Proceed to the next window: set n=n+1 , and increment δ by $\Delta\kappa + \Delta\nu$

9. Repeat from Step 4, until the full integer (coarse) delay range $\kappa=\kappa_0 \ldots K_v$ of the interpolating delay line is exhausted. (K_v is the maximum variable delay, κ_0 is the starting coarse delay value for the search)

10. Store $\delta_{\max}$ as δ_1 .

11. Read the programmed fine search step size $\Delta\nu$.

12. Repeat Steps 1 and 4 to 8 for the fine search: employ fractional-sample increments of$\Delta\nu$, covering the delays $\delta_1 - 1 \le \delta \le \delta_1 + 1$, ensuring not to exceed $0 \le \delta \le K_V$

13. Store $\delta_{\max}$ as δ_2 .

14. Set and freeze the interpolating delay line delay toδ_2 .

15. Apply the phase correction γ

[00057] Once the full course range is exhausted, the stored δ value $\delta_{\max}$ is used a s a starting point for a fine search. With $\Delta\ \kappa = 0$, the $\Delta\nu$ is incremented fractionally so as to search the delays $\delta_1 - 1 \le \delta \le \delta_1 + 1$. Once found (using an analogous loop to that used for the coarse search), the phase correction γ is found along with the proper delay value. It should be clear that $\gamma = \angle P_\delta$.

[00058] Updating of the phase LUT is only to be accomplished after the phase synchronization of the feedback signal. Freezing the phase LUT update can be done by setting $\mu F = 0$ and $\mu G = 0$.

[00059] After the phase correction is done, the magnitude of the feedback signal z(k) has to be adjusted as well. This adjustment is required to compensate for any gain in the system output signal relative to the input signal. Magnitude adjustment is accomplished by directly multiplying the feedback signal z(k) by a factor $A_{sx}(k)$. This is found using the following formula:

$$A_{sx}(k+1) = A_{sx}(k) + \mu_A \cdot (|x_\delta(k)| - A_{sx}(k) \cdot |z(k)|)$$

$$\text{For } k_A \leq k \leq k_{A+} W_A - 1$$

where

k_A = commencement time for the magnitude adjustments (preferably as early as possible)

W_A = number of samples for which the adjustments are performed

μ_A = update step size and $0 \leq \mu_A \leq 1$

$A_{sx}(k_A)$, μ_A, and W_A can be programmable.

[00060] Referring to Figure 6, a circuit for implementing the magnitude adjustment is illustrated. As can be seen, the magnitude 400 of the portion of the system output signal is multiplied with the factor $A_{sx}(k)$ by way of multiplier 410. The result is to be used in calculating the desired predistortion modification. This result is also subtracted from the magnitude ($|x_\delta(k)|$) of the delayed signal by way of adder 420. The result of the subtraction is multiplied by the update step size μ_A by way of multiplier 430 and added to the factor $A_{sx}(k)$ by adder 440 to produce the next value in the sequence for the factor. The delay element 450 delays the resulting value until it is ready for use.

[00061] Once the delay, magnitude and phase adjustments have been made, new equations for the error signals can now be used:

$$E_m(k) = A_{sx} \cdot |z(k)| - |x_\delta(k)|$$

$$E_\phi(k) = (\gamma + \angle z(k) - \angle x_\delta(k))$$

[00062] These two error signals can therefore be used in place of the earlier ones and these take into account the effects of the delay, magnitude, and phase differences between the system output signal and the input signal.

[00063] While the above embodiment utilizes LUT entries that are adjusted based on the difference between the input signal and the system output signal, the system output signal alone may be used, if desired, in adjusting the LUT entries. This alternative would require the modulation of the system output signal and the regeneration of a signal similar to the input signal. Successful recovery is possible if the amount of distortion is restricted such that demodulation can occur without errors. This approach is not desirable if the input signal is readily available to be used in determining the parameters of the adjustments. Another alternative involves measuring the distortion by measuring the amount of out-of-band energy. Adjustments can then be made to minimize the amount of out-of-band energy and, consequently, minimize the distortion.

[00064] Referring to Figure 3, a distortion monitor 500 is illustrated as receiving input from both update blocks 230, 260. The distortion monitor 500 monitors the distortion by way of the error functions defined above as:

$$e_M(k) = A_{Sx}\cdot|z(k)| - |x_\delta(k)|$$
$$e_\varphi(k) = (\gamma + \angle z(k) - \angle x_\delta(k))$$

where γ is the complex phase difference between x_δ (k) and z(k). By monitoring the distortion, the predistortion subsystem can detect LUT divergence. If such a condition is detected, a full or a partial re-initialization may be in order.

[00065] The error signals given above are weighted as:

$$\eta(k) = s\cdot|e_M(k)| + (1-s)|e_\varphi(k)|$$

with $0 \le s \le 1$. The averaging of $\eta(k)$ can be done using a leaky integrator

$$E(k) = \lambda\cdot\eta(k) + (1-\lambda)(\eta(k-1)$$

where the update parameter is λ.

[00066] The leaky integrator may be implemented as in the block diagram in Figure 6A where $\lambda = 2^{-n}$ which reduces the two multiplication operations into two simpler shift functions and one add function. This leaky integrator is a first order infinite impulse response low pass filter.

[00067] To control all the different components of the adaptive predistortion subsystem

200, a control block 600, controllably coupled to the different components, is provided. The control software run by the control block 600 for the above described predistortion subsystem allows for different actions to be taken for different conditions. The software can be viewed as a state machine with each state allowing the subsystem to execute specified actions. Thus, when the subsystem enters a specific state, it is equivalent to running a subroutine as specific actions and tasks are executed by the subsystem.

[00068] The control software and its analogous state machine can therefore be viewed as a control system that determines the actions of the predistortion subsystem. The various states and the steps taken when the predistortion subsystem enters it are listed and explained below:

State : Reset [RST]

The reset state resets all variables in the subsystem to their initial values and commences the synchronization and adaptation of the LUTs without regard to previous results or actions.

[00069] This reset state is activated through a reset external to the subsystem or through a self reset condition.

[00070] During a reset, all the LUT entries are reset to an all-zero state and the LUTs are not updated. Furthermore, in a reset state, the whole predistortion function of the subsystem 200 is disabled. This means that the magnitude value and the phase values of the input signal are not predistorted but are merely allowed to pass through the subsystem 200. The LUT entries can then by either loaded with preset values or be filled with all-zero entries.

[00071] Also during a reset, further adjustments to the phase and/or magnitude of the feedback signal are disabled while the delay lines are preset to their starting values. The feedback from the system output signal is disabled and the distortion monitor is reset to zero or to a preprogrammed value.

[00072] State: Pause [PSE]

During the Pause state, all the functions of the predistortion subsystem 200 are halted and all the settings, values, variables, and entries in the subsystem are frozen and feedback is disabled.

[00073] State: Hold [HLD]

The Hold state is analogous to the Pause state with the exception that the subsystem, except for the feedback, is functioning. The predistortion subsystem 200 freezes all the variables and LUT entries and turns off the feedback from the system output signal while still predistorting the phase and magnitude signals. However, the phase and magnitude LUT entries are not being adjusted.

[00074] State: Warm Up [WRM]

The Warm Up state is the same as the Pause state with the exception that the feedback from the system output signal enabled.

[00075] State: Update [UPD]

For the update state, the settings for the circuits which adjust the phase and magnitude of the feedback signal and the delay blocks 240, 270 are frozen while the rest of the subsystem 200 functions normally.

[00076] State: Full Initialization [FIN]

The full initialization state initializes the update and feedback circuitry. In this state, the LUT entries are frozen in that the update blocks 220, 260 are disabled while the feedback path is active along with the additive adjustments of the phase and magnitude values. However, the distortion monitor control value and update duty control are both reset along with the update speed control (to be discussed below).

[00077] State: Re-Initialization [RIN]

For re-initialization, the feedback gain and phase adjustments are adjusted while the delay settings (for the delay blocks 240, 270) are not modified. For this state, the phase and magnitude update blocks are disabled but predistortion continues. The distortion monitor control value is reset while the gain and phase adjustments to the feedback signal are still operative.

[00078] State: Cool Down [CLD]

The cool down state maintains the feedback function after an LUT update to avoid problems which may be caused by the feedback circuit. It initiates a cool down timer, the duration of which is user configurable to enable avoidance of the feedback related problems. During this state, the distortion monitor is disabled (frozen) and the other relevant settings, such as the LUT entries, the setting for the feedback gain adjustment, the delay blocks 240, 270, and the feedback phase adjustment circuitry, are also frozen. More importantly, the phase and magnitude LUTs are also disabled.

[00079] The state transitions which the control block 600 undergoes between the different states is detailed in the table below. To assist in the understanding of the transition table below, Figure 6B is presented. Figure 6B is a state transition diagram illustrating the different states and the sequence of state transitions possible. For clarity, a listing of the timers implemented by the block 600 is also presented below.

State Transition Table

From	To	Condition
any	RST	Reset externally applied to subsystem
RST	PSE	Completion of RST operations
PSE	WRM	Transmit process enable and update request
	HLD	Transmit process enable otherwise
WRM	SBY	Start of transmit packet samples at predistortion subsystem
SBY	FIN	Expiration of SBY timer and 1. First SBY since last RST or 2. Full initialization procedure incomplete since last FIN or 3. Full initialization request

	RIN	Expiration of SBY timer, and SBY to FIN conditions do not apply, and 1. Reinitialization procedure incomplete since last RIN or 2. Reinitialization request
	UPD	Expiration of SBY timer, otherwise
UPD	CLD	End of transmit packet samples at predistortion subsystem
CLD	RST	Expiration of CLD timer and Reset request
	PSE	Expiration of CLD timer otherwise
HLD	PSE	End of transmit packet samples at predistortion subsystem
FIN	CLD	End of transmit packet samples at predistortion subsystem
	UPD	Otherwise
RIN	CLD	End of transmit sample packets at predistortion subsystem
	UPD	Othewise

Timers Implemented by Control Block

Configuration Parameter	Description	Range
SBY-Timer	Timer duration	0,2,4,8,16,32,64, or 128 samples
CLD-Timer	Timer duration	0,2,4,8,16,32,64, or 128 samples
RTO-Timer	Recency time-out timer	0..1 s in steps of at least 1 ms and ‘inf’

[00080] To build in more flexibility and robustness in the predistortion subsystem 200, the update speed parameter μ_G and μ_F are, as noted above, configurable. Another lookup table, internal to the control block 600, provides 4 possible settings for each of the update speed parameters. These possible values and the time slot (expressed in terms of the number of symbols received) for which they would be used are listed in the table below.

μ_F	μ_G	Time slot (expressed in a symbol count, during which the LUTs are updated)
1/2	1/4	0..15
1/4	1/8	16..47
1/8	1/8	48..79
1/16	1/8	81 and beyond

[00081] As noted above, an update duty control (a control value) is also used by the control block 600. The update duty control is a value which determines how often an LUT update is to be performed. In one contemplated application of the invention, the data in the signal to be predistorted arrives in discrete packages or packets with each packet containing multiple symbols. The LUT update duty cycle is expressed in terms of a fraction of the number of packets during which updates are performed. For instance, at a duty cycle of 1 the system updates the LUT's during all transmit packets, at 1/9 the system updates during the first packet, and then freezes the LUT during the following 8 packets: transmission and pre-distortion then occurs by means of the Hold state. In this case the cycle is 9 packets, the duty is one packet, and the idle time is 8 packets. More precisely, the duty cycle is defined by the idle time, and after expiration of the idle time an update request is issued.

[00082] In this application of the invention , the duty cycle is defined by the idle time with a request for an LUT update being made after the idle time has expired. The duty cycle may be defined by other criteria such as the number of packets received but, for this application, it was found that the idle time was a more convenient choice.

[00083] It should however be noted that the duty cycle is not operative at all times. It is only applied after a specified delay of a specified number of symbols has been received. Before the symbol count, an LUT update is requested for each packet. A reset of the update duty control value accomplishes a start of a new period starting with a duty and a start of the delay symbol count. The duty is therefore 100% for an initial number of symbols after a reset of the update duty control value and, subsequently, the duty is as configured.

[00084] The table below documents the different possible values for the delay symbol count configuration.

Configuration Parameter	Description	Range
UDC-IDL	Idling time between update requests	0, 2^1 ... 2^6, and 'inf' packets (8 possible values)
UDC-DLY	Delay in terms of a transmit symbol count after which the duty cycle applies	0, 2^1 ... 2^6, and 'inf' symbols (8 possible values)

[00085] For clarity, a value of "inf" for the idle time denotes no duty (no LUT updates requested as ever) and a value of 0 for the idle time denotes LUT updates for each packet. A delay value of 0 denotes that the duty cycle is effective immediately.

[00086] Another control value used by the control block 600 is the initialization duty control. This initialization duty control value control how many initializations are performed for every update of the LUTs. The initialization duty cycle is expressed in terms of a fraction of the number of update packets at the start of which an initialization is performed. For instance, at a duty cycle of 1 the system performs an initialization at the start of all update transmit packets, at

1/9 the system initializes at the first update packet, and performs an update without initialization during the following 8 update packets. In this case the cycle is 9 update, the duty is one update, and the idle time is 8 updates.

[00087] The distortion monitor 500 may, depending on the implementation, be a simple component that issues an alarm if the distortion exceeds a certain preset distortion value. However, a more complex yet more useful and robust implementation is possible. Such an implementation may use multiple states and state transitions for the distortion monitor. These states can be as follows:

Reset (resets the monitor)

Measure (monitors the distortion metric and requests a predistortion subsystem recovery of the distortion metic exceeds the preprogrammed threshold)

Freeze (holds the state of the distortion metric between packets or during initialization or re-initialization of the predistortion subsystem)

No Monitor (disables the alarms and does not monitor the distortion metric. This allows the predistortion subsystem to converge and the distortion to settle)

Monitor (compares the distortion metric with the programmed threshold and enables the triggers to be tripped if the threshold is exceeded)

[00088] Different alarm levels and states may also be implemented such that, depending on the alarm level and state, different actions may be taken to address the distortion (e.g. full initialization of the subsystem, a reset of the subsystem, a re-initialization etc.).

[00089] To execute and implement the states noted above and to execute the software, the control block 600 may be implemented as a general purpose central processing unit with attendant circuitry. Referring to Figure 6C, a block diagram of such a control block 600 is illustrated. The CPU 610 is coupled to some random access memory (RAM) 620, a register file 630, and read only memory (ROM) 640. The CPU 610 performs the execution of the software and interfaces with the predistortion subsystem 200 while the registers 630 and the RAM 620 hold the temporary values (such as the various control values) required by the software in the CPU 610. The ROM 640 holds the software being executed by the CPU 610.

[00090] While the control block 600 above is illustrated as a general purpose data processing system, other implementations, such as a application specific integrated circuit (ASIC) or combinational logic circuit in which the logic of the software is hardwired in the hardware, are possible.

[00091] While the above embodiment utilizes LUT entries that are adjusted based on the difference between the input signal and the system output signal, the system output signal alone may be used, if desired, in adjusting the LUT entries. This alternative would require the modulation of the system output signal and the regeneration of a signal similar to the input signal. Successful recovery is possible if the amount of distortion is restricted such that demodulation can occur without errors. This approach is not desirable if the input signal is readily available to be used in determining the parameters of the adjustments. Another alternative involves measuring the distortion by measuring the amount of out-of-band energy. Adjustments can then be made to minimize the amount of out-of-band energy and, consequently, minimize the distortion.

[00092] Regarding the amplifier subsystem 10, Figure 7 illustrates one embodiment of the subsystem 10. In Figure 7, the signal decomposer 20 of Figure 1 comprises a phasor fragmentation engine 20A along with phase modulation units 60A, 60B. The fragmentation engine 20A receives the magnitude (M) and phase (ϕ) representing the predistorted signal. The phasor fragmentation engine 20A deconstructs a predetermined modulation waveform (the predistorted signal) into signal components which are of equal and constant magnitude. Further information regarding the phasor fragmentation engines may be found in the applicant's co-pending application US application No. 10/205,743 entitled COMPUTATIONAL CIRCUITS AND METHODS FOR PROCESSING MODULATED SIGNALS HAVING NON-CONSTANT ENVELOPES , which is hereby incorporated by reference. In Figure 7, these signal components are denoted by angles α and β. These components are each received by RF modulation and filtering blocks 60A, 60B which process the components to produce RF modulated and filtered versions of the components. The signal component 70A is an RF signal with phase α while signal component 70B is an RF signal with phase β. These components 70A, 70B are then amplified by amplifiers 90A, 90B. The amplified components are then recombined using

combiner 100. It should be noted that the phase modulation, also known as carrier modulation, may also introduce some undesired amplitude modulation. Signal decomposition methods other than the phasor fragmentation referred to above may also be used by the signal decomposer 20.

[00093] Regarding the Chireix architecture amplifier subsystem 10, it has been found that, for higher amplification efficiencies, switch mode amplifiers are preferred for the amplifiers 90A, 90B. Such switch mode amplifiers, specifically Class D and Class F power amplifiers, provide low output impedances that allow higher amplification efficiencies. A co-pending application filed on October 16, 2002 and having US Serial No. 10/272,725 entitled CHIREIX ARCHITECTURE USING LOW IMPEDANCE AMPLIFIERS provides further information on the desirable components and is hereby incorporated by reference. Such types of amplifiers are not required for the invention to function but they have been found to provide performance at a desirable level.

[00094] It should further be noted that while those are only two parallel amplifiers 90A, 90B in Figure 1 and Figure 7, multiple parallel amplifiers may be used as long as the decomposer 20 decomposes the predistorted signal 130 into enough components so that each component is separately amplified and phase modulated in parallel with the other components.

[00095] It should also be noted that the predistortion subsystem 10 explained above does not linearize a power amplifier as is well-known in the field. Instead, the predistortion subsystem linearizes a whole power amplifier system - the output of the whole amplifier system is linearized and not simply the output of a single amplifier. Also, unlike the linearizing systems for power amplifiers that are currently known, the amplifier system discussed in this document compensates for distortions that mostly occur at mid signal amplitudes. Current single amplifier linearization systems linearize distortions that occur at large signal amplitudes.

[00096] It should further be noted that the invention may be applied to any signal processing system which decomposes a signal into components and recombines them. It has been found that signal combiners (block 100 in Figure 1) invariably cause distortions. These combiners use addition to recombine the components and improper signal addition, such as when recombining sinusoidal components, has been found to be one cause of the distortions in the system output signal. In the above embodiment, the phasor fragmentation engine decomposes

the incoming signal into vectors and the improper addition of these vectors by the combiner 100 lead to distortions in the output signal.

[00097] While the above embodiment amplifies the input signal, albeit separately for each component, this need not be the only signal processing accomplished after the input signal is decomposed. Referring to Figure 8, such a generalized system (which may be part of a larger signal transmission system) is illustrated. The predistortion subsystem 120 predistorts an incoming signal 30 and compensates for distortions introduced in the system output signal 110 by the improper or imperfect recombining of the input signals components. These components are produced by the signal decomposer 20 and are separately processed by signal component processor blocks 75A, 75B. The processing executed by the blocks 75A, 75B may take the form of amplification (as in the embodiment above), phase modulation, a combination of the two, or any other signal processing which may be desired. As an example, each of the signal components illustrated in Figure 7 may be separately phase modulated in addition to being amplified by amplifiers 90A-90B. The phase modulation may be accomplished separately or be incorporated in the signal decomposer or, as contemplated for the implementation illustrated in Figure 7, incorporated into the modulation and filtering blocks 60A, 60B.

[00098] As can be seen in Figure 8, the signal processing subsystem 10A receives the predistorted signal from the predistortion subsystem 120. After being received, the predistorted signal is decomposed by the signal decomposer 20 into components. These components are then separately processed by the signal component processor blocks 75A, 75B and are then recombined by the recombiner 100.

[00099] A feedback signal processing block 400 receives a portion of the system output signal 110 and processes this portion so it may be used by the adaptive predistortion subsystem 120. As an example, the feedback signal processing block 400 may contain the A/D converter 330 and the conversion unit 340 illustrated in Figure 4. The magnitude adjustment and phase adjustment detailed above can also be implemented in this block 400.

[00100] One advantage using the above invention is that it allows less stringent tolerances to be used for the system components. Previously, components had to be substantially matched so that signal processing could produce acceptable results. By using the above invention, less

than substantially matched components may be used together. Errors due to a mismatch may be measured and compensated for by the predistortion subsystem.

[00101] Referring to Figure 9, a detailed block diagram of a system incorporating the features illustrated in Figure 3, 4 and 8 is presented. As can be seen, the adaptive predistortion block 120 in Figure 8 is comprised of the separate magnitude delay 240 and phase delay 270 along with a magnitude predistortion calculation block 260A. The inputs of these calculation blocks 230A, 260A are the delayed input signals from the delay blocks 240, 270 and the adjusted feedback signals from the magnitude adjustment blocks 410 and the phase adjustment block 420. After the magnitude and phase predistortion modification are calculated, then the magnitude LUT block 220A and the phase LUT block 250A apply the predistortions. It should be noted that the magnitude LUT block 220A includes the magnitude LUT 220 and the magnitude LUT update block 230 shown in Figure 3. Similarly, the phase LUT block 250A incorporates the phase LUT 250 and the phase LUT update block 260 illustrated in Figure 3.

[00102] As can also be seen, the feedback signal processing block 400 illustrated in Figure 8 is comprised of the Cartesian to polar coordinate conversion block 340 which feeds the magnitude adjustment block 410 and the phase adjustment block 420.

[00103] A person understanding this invention may now conceive of alternative structures and embodiments or variations of the above all of which are intended to fall within the scope of the invention as defined in the claims that follow.